A More Implementation Details

More Training Details. We use a batch size of 12, with each batch containing one training scene consisting of input views and target views. As training progresses, the frame distance between input views gradually increases. The initial learning rate is set to 1×10^{-5} for the backbone and 1×10^{-4} for other parameters.

More Details of Pose Head. The pose head structure is shown in Fig. 6. During training, the linear layer for rotation is initialized with zero weights, and the 6D bias is set to $(1, 0, 0, 0, 1, 0)$ to approximate the identity matrix, ensuring that the initial pose for each view has a shared field of view for stable convergence. Camera normalization sets the pose of the first view to the identity matrix.

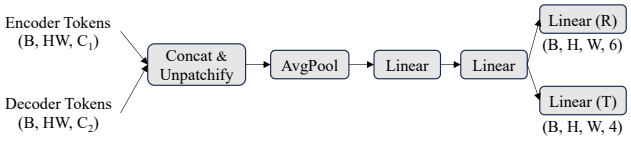

Figure 6. The structure of the proposed pose head.

More Details of Splatt3R Results. In Tab. 1 and Tab. 2 of the main paper, we retrain Splatt3R since its original implementation does not provide results on RE10K and ACID. We remove the loss mask, as it depends on ground-truth depth, which is unavailable for RE10K. Additionally, the original implementation employs an offset head to adjust the 3D points predicted by the frozen MASt3R. However, we find that this approach inefficient for aligning with the scale of ground-truth intrinsics. Instead, we directly fine-tune the 3D point head while keeping the backbone frozen. Training is conducted using ground-truth poses.

More Details of Baselines in Pose Estimation. In Tab.3, all experiments are conducted using 256×256 input images. For SuperPoint + SuperGlue, feature matches are used to estimate Essential Matrices and compute relative poses. For DUSt3R and MASt3R, we estimate camera intrinsics from the 3D points of the first view and compute relative poses using the PnP algorithm [15]. Since SelfSplat defines the target image as the reference frame, we evaluate relative poses from the target to the context image, a relatively easier task than predicting relative poses between two context views. NoPoSplat estimates poses in two stages: it first initializes the relative pose between input views using PnP [15] with RANSAC [13], leveraging predicted Gaussian centers. Then, with the Gaussian parameters fixed, it refines the pose by minimizing photometric losses combined with an SSIM loss. This second-stage optimization integrates 3D Gaussian splatting into the loop, making it computationally expensive and less suitable for real-time applications. For a fair comparison with other splatting-based methods, we report NoPoSplat's accuracy based on the first-stage initialization only.

More Details of Ablation on Ground-truth Poses. To evaluate our method's ability to reconstruct geometry without pose supervision, as shown in Tab. 5, we incorporate camera poses to supervise our pose head. Our pose loss is a combination of geodesic loss [34] for rotation and L_2 distance loss for translation. Specifically, they are defined in Eq. 6 and Eq. 7. We set the weight for rotation loss to 0.1, the weight for translation loss to 0.01.

$$\mathcal{L}_{\text{rot}} = \arccos\left(\frac{\text{trace}(\hat{\boldsymbol{R}}^T \boldsymbol{R}) - 1}{2}\right) \tag{6}$$

$$\mathcal{L}_{\text{trans}}(\hat{\boldsymbol{T}}, \boldsymbol{T}) = \|\hat{\boldsymbol{T}} - \boldsymbol{T}\|_2^2 \tag{7}$$

B More Experimental Analysis

Evaluation on In-the-Wild Data. We evaluate our model on mobile phone photos using the version trained without intrinsic embeddings (as in the ablation study). Given two input images, we estimate the focal length from the output Gaussian centers of the canonical view (the first image). This focal length is used to render novel views. The 3D geometry and rendered results in Fig. 7 demonstrate our model's strong out-of-domain generalization, even under large viewpoint changes.

Figure 7. 3D Gaussians and rendered RGB and depth results on mobile phone photos.

Evaluation on the Evaluation Set of pixelSplat. We adopt the evaluation set from NoPoSplat [49] in our main paper, as it presents a greater challenge due to minimal overlap between most input pairs. Additionally, we report results using the evaluation sets from pixelSplat [4] and MVSplat [8], as shown in Tab. 8.These results also demonstrates that our method consistently outperforms other SOTA methods.

Comparison on PnP Pose and Pose Head. Our method supports two strategies for pose estimation: direct regression via the pose head, and estimation via PnP [15] with RANSAC [13], using the predicted 3D Gaussian centers. As shown in Tab. 9, both approaches achieve comparable results in both in-domain and out-of-domain settings, indicating strong alignment between the estimated poses and the predicted Gaussian centers.

Comparison with MASt3R. As our method is initialized from MASt3R weights, to compare with the original

Method	RE10K			ACID		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
pixelSplat	26.090	0.863	0.136	28.270	0.843	0.146
MVSplat	26.387	0.869	0.128	28.254	0.843	0.144
NopoSplat*	26.820	0.880	0.125	27.952	0.837	0.150
Ours	27.311	0.888	0.119	28.407	0.845	0.142
Ours*	**27.696**	**0.892**	**0.117**	**29.000**	**0.862**	**0.136**

Table 8. Novel view synthesis performance comparison on the evaluation sets of pixelSplat and MVSplat.

Task	Method	Rotation			Translation		
		5° ↑	10° ↑	20° ↑	5° ↑	10° ↑	20°
RE10K → RE10K	PnP	0.793	0.875	0.926	0.661	0.789	0.872
	Pose Head	0.816	0.886	0.932	0.666	0.793	0.874
RE10K → ACID	PnP	0.614	0.739	0.830	0.384	0.545	0.683
	Pose Head	0.645	0.754	0.838	0.402	0.555	0.689

Table 9. Comparison between PnP poses estimated from Gaussian centers and poses predicted by the pose head.

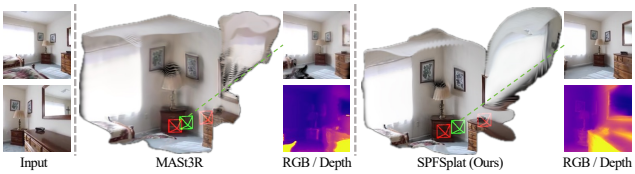

Figure 8. Comparison of 3D Gaussian and rendered results between MASt3R and our method.

Method	PSNR↑	SSIM↑	LPIPS↓
MASt3R	17.617	0.539	0.403
SPFSplat (Ours)	**25.484**	**0.847**	**0.153**

Table 10. Novel view synthesis performance comparison between MASt3R and our SPFSplat on RE10K.

MASt3R, we adapt MASt3R for novel view synthesis by freezing its weights, setting its output 3D points as Gaussian centers, and adding a DPT head to predict other Gaussian parameters (as in our approach). To address scale inconsistency, we estimate the focal length from the 3D points and use it for training instead of the ground-truth focal length. The model is trained with ground-truth poses. As shown in Tab. 10 and Fig. 8, although our model is initialized from MASt3R and trained without pose supervision, it significantly outperforms MASt3R, achieving more accurate geometric structures and visual details.

Initialization	PSNR↑	SSIM↑	LPIPS↓
Random	21.200	0.690	0.250
DUSt3R	25.280	0.841	0.156
MASt3R	**25.484**	**0.847**	**0.153**

Table 11. Comparison of different initialization strategies

Initialization. In our main paper, we initialize the backbone with MASt3R weights. Here, we further analyze the influence of different backbone initialization strategies. As shown in Tab. 11, MASt3R's pretrained weights achieve slightly better NVS performance compared to DUSt3R. This improvement can be attributed to MASt3R's training on feature-matching tasks, which produces stronger local feature representations that improve both pose estimation accuracy and the quality of reconstructed 3D Gaussians. For random initialization, we adopt a warm-up phase by incorporating a point cloud distillation loss from the DUSt3R model during the first 10,000 steps. This additional supervision is crucial, as training with only photometric loss, especially without ground-truth geometric supervision, makes it difficult for the network to learn to predict Gaussians in the canonical space. Since our model is trained without ground-truth poses, proper initialization significantly improves pose estimation quality. Although random initialization results in a noticeable performance drop, the results still demonstrate the model's capability to reconstruct Gaussians without known poses.

Method	NVS			Pose		
	PSNR↑	SSIM↑	LPIPS↓	5° ↑	10° ↑	20° ↑
(a) Ours	**25.484**	**0.847**	0.153	0.617	0.755	0.845
(b) w/o intrin. emb.	24.864	0.829	0.161	0.562	0.717	0.823
(c) w/o reproj. loss	19.836	0.644	0.289	0.028	0.102	0.263
(d) w/ gt pose loss	25.239	0.842	0.157	**0.691**	**0.810**	**0.885**
(e) w/o L_2 loss	24.310	0.837	**0.150**	0.602	0.740	0.833
(f) w/o LPIPS loss	25.336	0.832	0.210	0.559	0.709	0.812

Table 12. Component ablations on RE10K. NVS are evaluated using predicted pose.

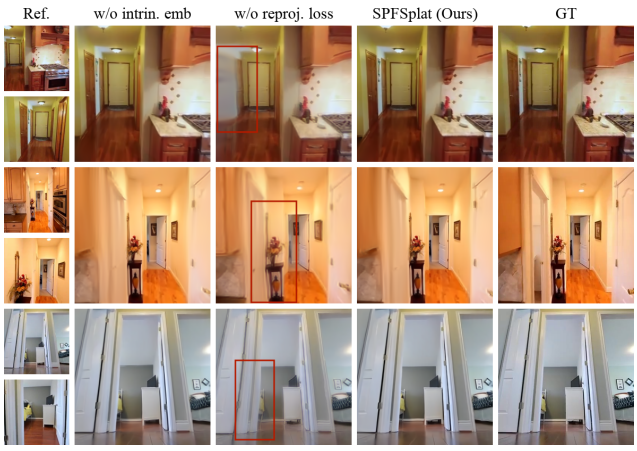

Figure 9. Ablation on the intrinsics embedding and reprojection loss. Some failure regions are highlighted by red rectangles.

More Ablation Results. We demonstrate the ablation results on RE10K evaluated using predicted poses in Tab. 12.

Figure 10. Qualitative comparison on different numbers of input views.

It indicates that the intrinsics embedding and reprojection loss both contribute to better alignment between the poses and Gaussians. We also incorporate the ablation of L2 and LPIPS in (e) and (f), which demonstrates that both L2 and LPIPS losses positively impact NVS and pose estimation. Fig. 9 presents the rendered results of our method without intrinsics embedding or reprojection loss. Removing intrinsics embedding leads to slightly blurrier outputs due to scale ambiguity. The absence of reprojection loss, however, results in severe blurring and rendering artifacts, as the lack of geometric constraints hinders proper alignment between poses and 3D points, causing reconstruction errors.

Extension to Multiple Views. Our method can be extended to multi-view input. For a fair comparison, we fix the first and last views across all experiments while gradually increasing the number of intermediate views. As shown in Fig. 10, more input views progressively enhance scene completeness and visual details, leading to improved rendering quality.

Failure Cases. Fig. 11 shows that our method may produce blurred outputs or artifacts in occluded or texture-less regions, or under extreme viewpoint changes. These issues may require generative abilities or explicit 3D supervision.

Figure 11. Some bad cases of our SPFSplat.

C More Visualizations

We show more qualitative comparisons with baselines in Fig. 12 to Fig. 15. Our method achieves stable and superior performance across different levels of image overlap and diverse datasets..

D Limitations

Our method can be trained without ground-truth poses and easily scales to large datasets, therefore, future work could explore training on larger, more diverse datasets to improve pose estimation and generalization ability. Moreover, since our approach is not generative, it cannot reconstruct unseen areas with detailed textures. Generative models could be leveraged to mitigate this limitation.

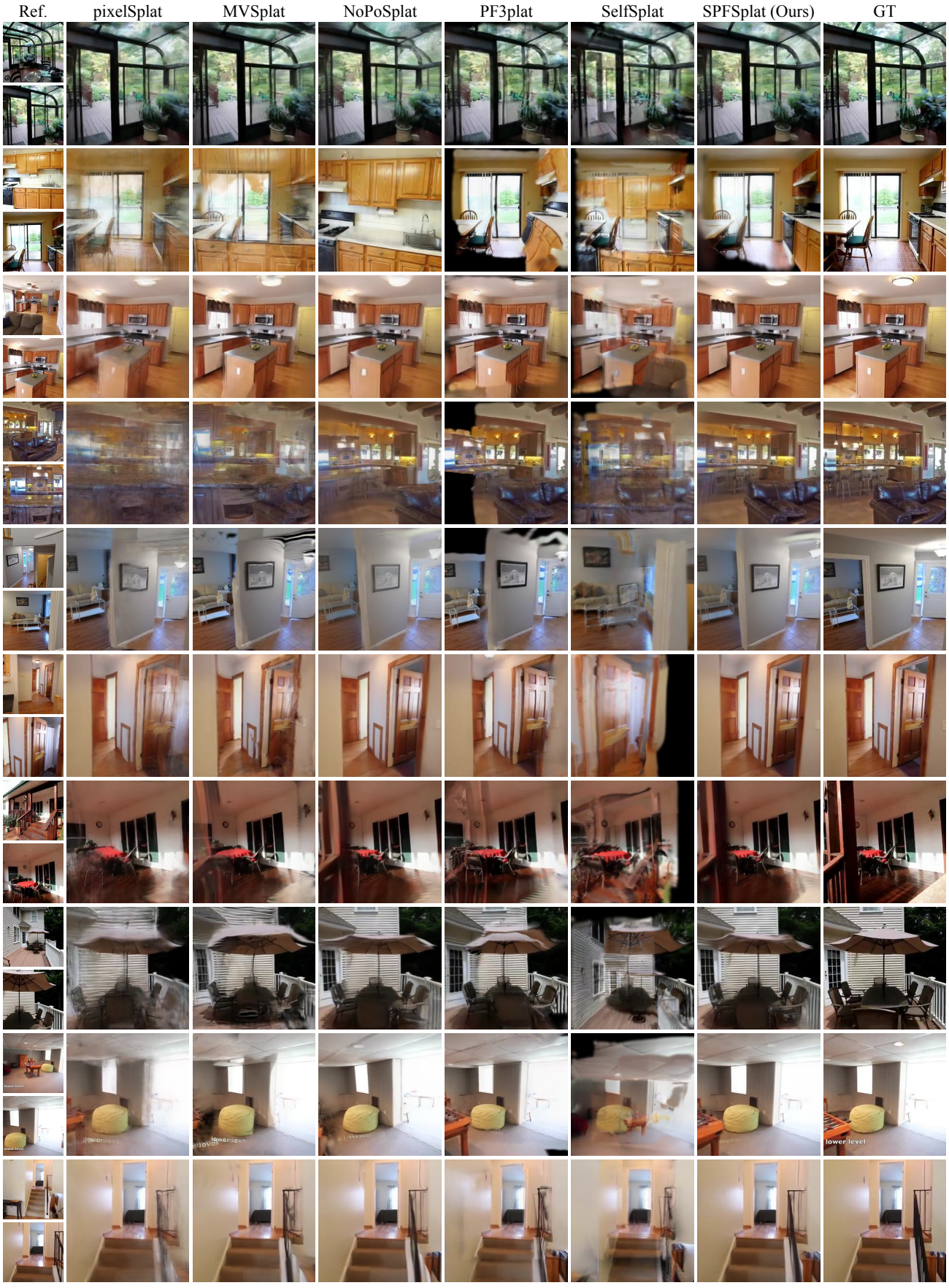

Figure 12. More qualitative comparisons on RE10K with small image overlap.

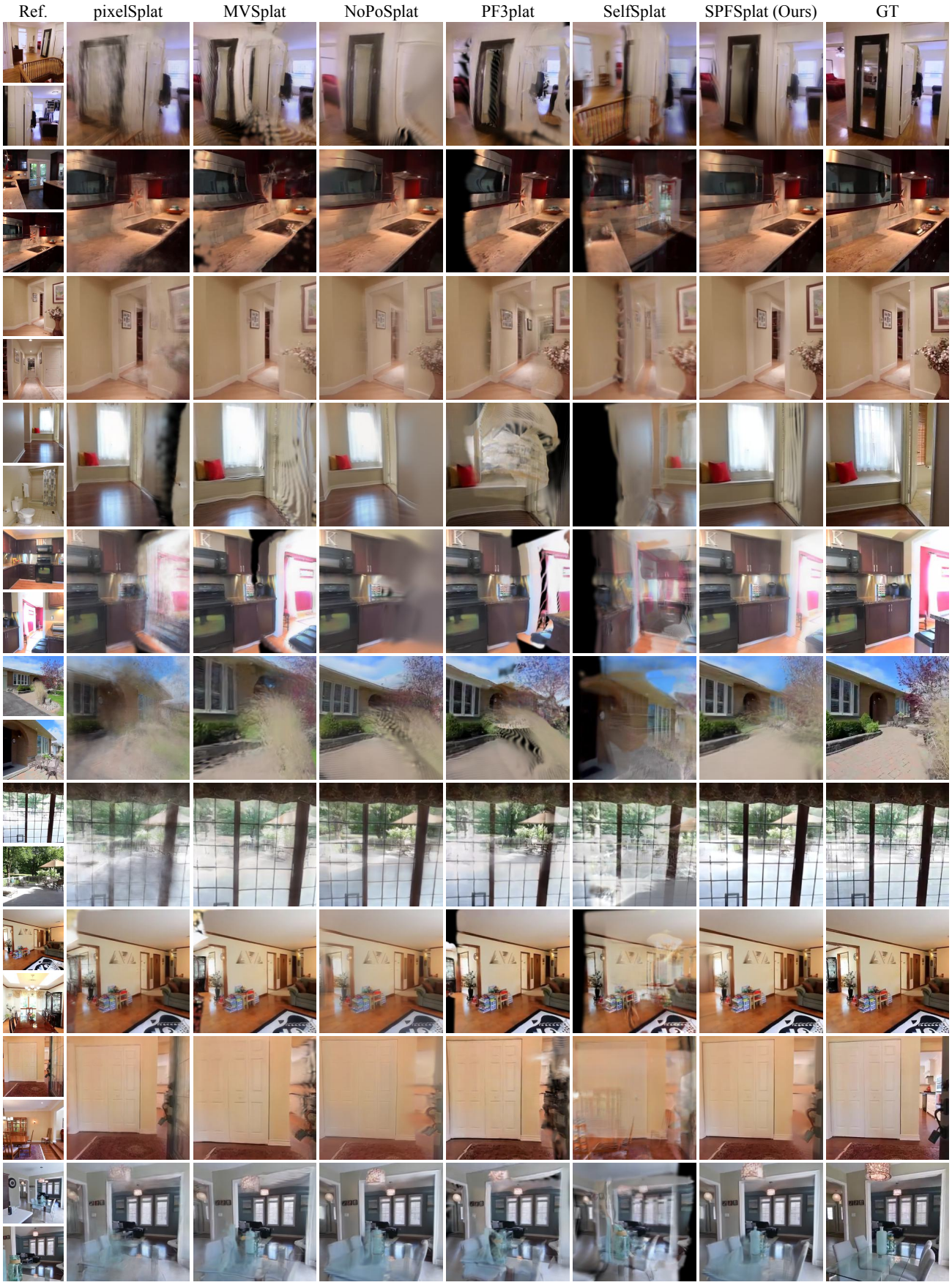

Figure 13. More qualitative comparisons on RE10K with medium image overlap.

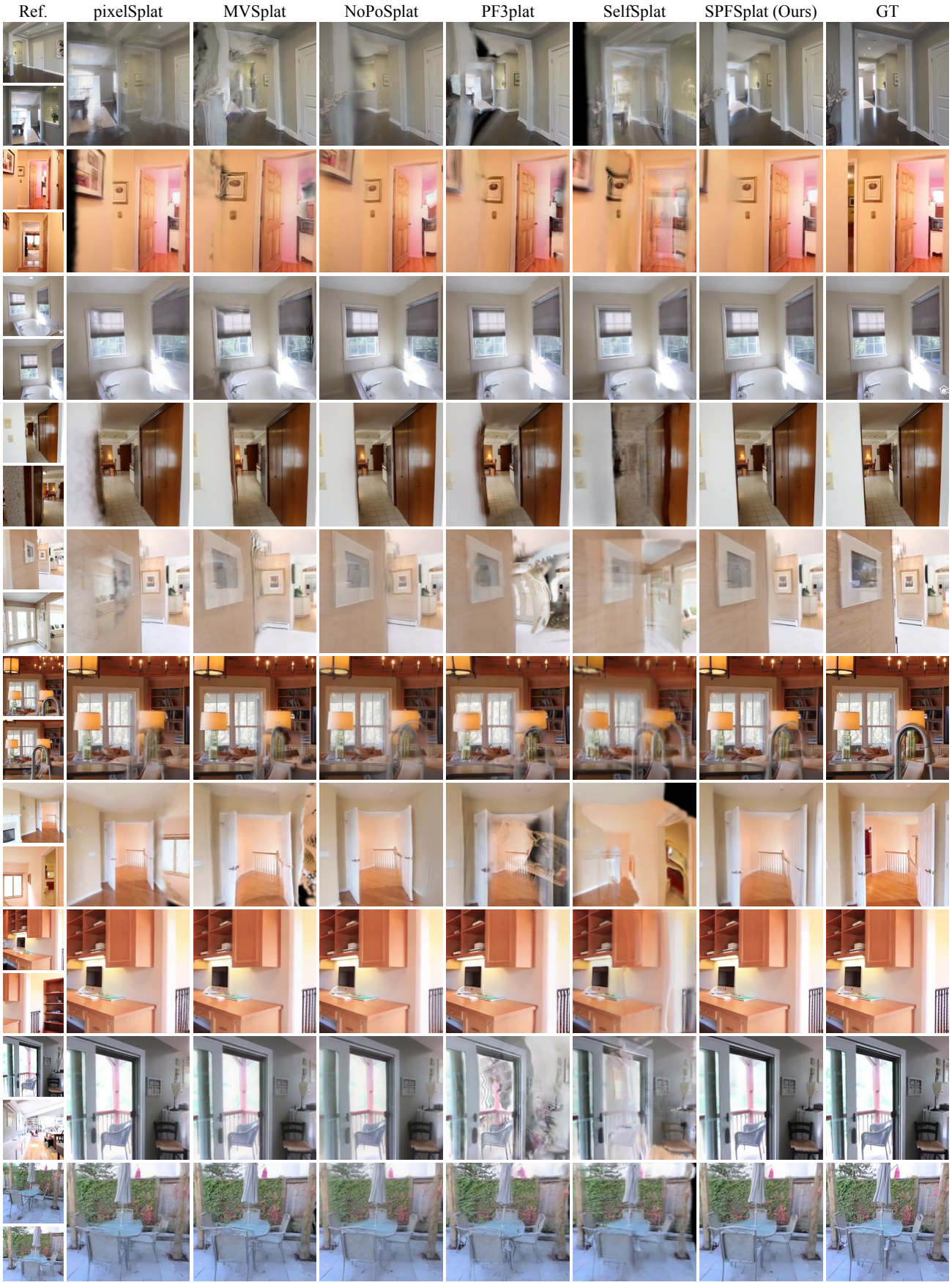

Figure 14. More qualitative comparisons on RE10K with large image overlap.

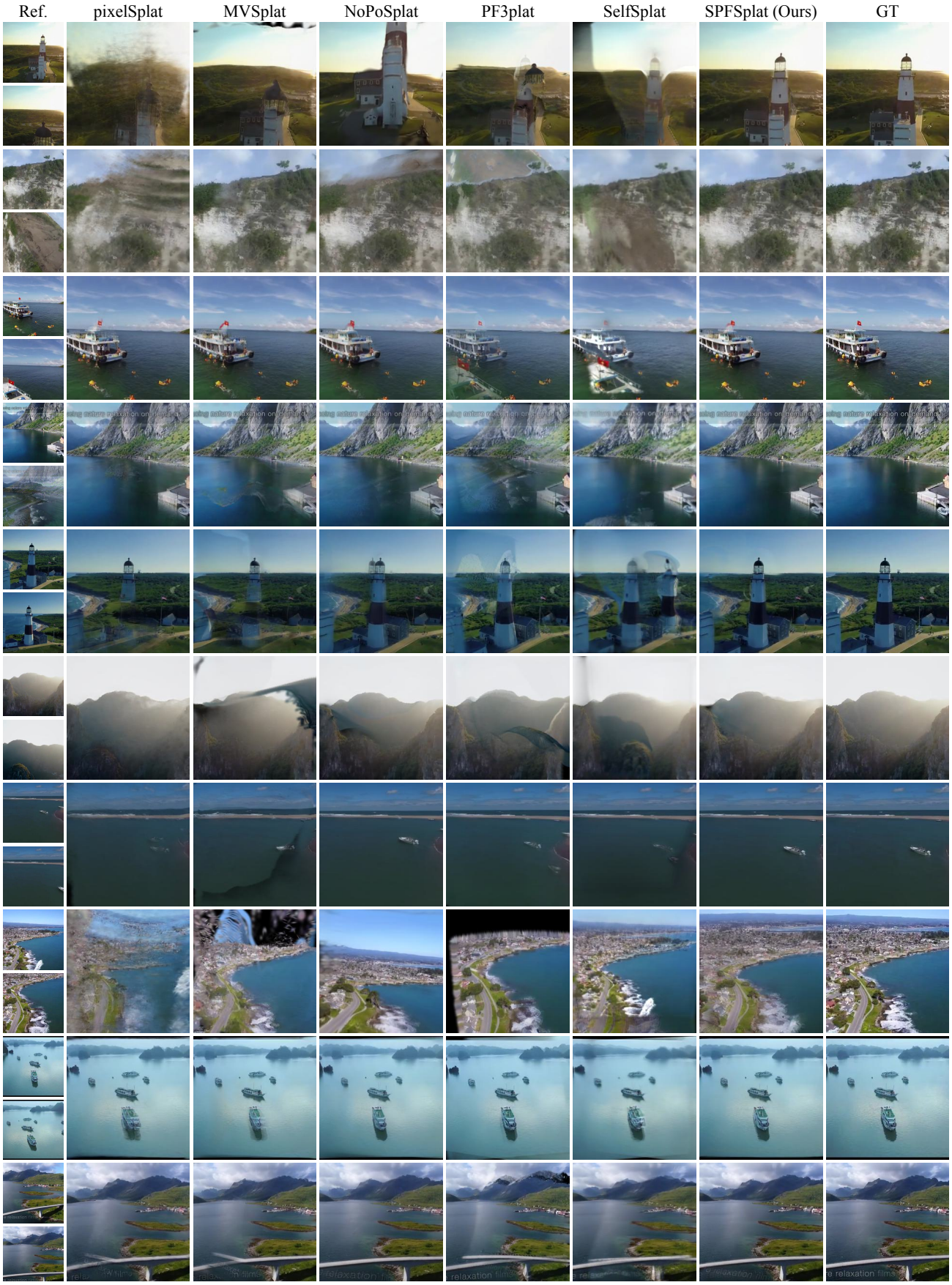

Figure 15. More qualitative comparisons on ACID.